

# 搭建织女星开发板的RISC-V开发环境

## 1. 基本介绍

本指南详细地介绍了如何给织女星开发板（RV32-VEGA-Lite）搭建一个 RISC-V 开发环境。织女星开发板是 OPEN-ISA 社区为中国大陆地区定制的一款体积小、功耗超低和功能丰富的基于 RV32M1 的 RISC-V 评估开发板。利用该开发板，用户可以快速建立一个使用 RV32M1 的 RISC-V 应用和演示系统。

RV32M1 是一颗真正的 RISC-V 芯片，其目的是为了扩大和推动 RISC-V 生态系统的发展，让广大的 MCU 嵌入式开发工程师有真正的 RISC-V 芯片可用。目前该芯片只供应 OPEN-ISA 社区生产评估开发板使用。RV32M1 集成了两个 RISC-V 核：一个 RISCY 核和一个 ZERO\_RISCY 核。两个 RISC-V 核供共用一个 JTAG 调试口。关于 RV32M1 芯片的详细信息，请参阅 RV32M1 的技术手册。

给织女星开发板搭建 RISC-V 开发环境包括安装 GNU GCC 工具链、OpenOCD 调试软件、Eclipse 集成开发环境和 Eclipse 针对 RISC-V 开发所需的相关配置、RV32M1 软件开发包（SDK）。RV32M1 的 RISC-V 开发环境支持 Windows、Linux 和 Mac 操作系统。在每个操作系统下，你根据自己的偏好和习惯，可以选择基于图形界面的 Eclipse 集成开发环境或者简单地使用命令行方式进行开发和调试。

根据你主机上的操作系统，选择下面其中一个安装步骤来搭建你的 RISC-V 开发环境。

- Windows
- Linux
- macOS

### 目录

1. 基本介绍 .....	1
2. Windows 下开发环境的搭建 .....	2
2.1 安装 Telnet 远程登陆客户端应用 .....	2
2.2 安装 GNU Eclipse IDE for C/C++ Developers .....	2
2.3 安装 Windows Build Tools .....	2
2.4 安装 RV32M1 GNU GCC 工具链 .....	3
2.5 安装 OpenOCD 调试软件和 J-Link 驱动 .....	4
2.6 安装 RV32M1 SDK .....	6
2.7 安装 CMake .....	6
2.8 为命令行方式安装 MinGW .....	7
3. Linux 下开发环境的搭建 .....	8
3.1 安装 RV32M1 GNU GCC 工具链和 CMake .....	8
3.2 安装 OpenOCD 调试软件 .....	9
3.3 安装 Eclipse IDE for C/C++ Developers .....	9
4. MacOS 下开发环境的搭建 .....	10
4.1 安装 RV32M1 GNU GCC 工具链和 CMake .....	10
4.2 安装 OpenOCD .....	11
4.3 安装 Eclipse IDE for C/C++ Developers .....	11
5 参考文档 .....	12
6 修订历史 .....	12



## 2. Windows下开发环境的搭建

本节详细描述在 Windows 环境下如何给 RV32M1 搭建 RISC-V 开发环境。开发环境包括 Eclipse 集成开发环境和命令行开发环境，有的安装步骤在特定的开发环境可能不是必须的。为了方便大家学习和使用，建议大家按照下面的步骤来完成开发环境的搭建。

### 2.1 安装Telnet远程登陆客户端应用

用户可以通过 telnet 终端连接 OpenOCD 调试软件来调试运行在 RV32M1 上的应用程序。如果想使用 OpenOCD + telnet 的方式来调试运行应用程序，要确保 Windows 系统的 telnet 客户端应用被安装了。特别是修改 RV32M1 的多核启动配置，必须要用 OpenOCD + telnet 的方式来修改。

在 Windows 的 cmd 控制台中，运行下面命令来安装 telnet 客户端应用：

```
pkgmgr /iu:"TelnetClient"
```

### 2.2 安装GNU Eclipse IDE for C/C++ Developers

安装 Eclipse IDE for C/C++ developers 的步骤如下：

1. 从 [Github](#) 下载 GNU MCU Eclipse IDE for C/C++。
  - GNU MCU Eclipse IDE for C/C++ 包括了 Eclipse CDT 和 MCU plug-ins。
  - 选择最新版的下载。目前最新版的不再支持 32 位操作系统。
  - 选择文件名后面带有 `-win32.win32.x86_64.zip` 字样的是 Windows 版本的安装包
2. 解压缩下载的安装包到用户指定的位置，比如 D 盘的根目录下，其解压后的路径为 D:\eclipse。

### 2.3 安装Windows Build Tools

这步不是必须的。在 Eclipse 工程可以选择 build 类型：外部 builder 还是内部的 builder。如果用户从不使用 Eclipse 外部 builder，可以跳过这一步。否则必须安装 GNU MCU Eclipse Windows Build Tools。Windows Build Tools 包括了调用交叉编译工具链所需要的 `make` 和 `rm` 等最小的可执行文件集合。因为外部 build 工具需要有 `make` 和 `rm` 这两个可执行文件。

在 Windows 系统下按照下面的步骤安装 Windows Build Tools：

1. 从 [Github](#) 下载 GNU MCU windows Build Tools。
  - 选择最新版的下载。
  - 32 位 Windows 下载文件名带有 `-win32` 的压缩包文件，64 位 Windows 下载文件名带有 `-win64` 的压缩包文件。

- 将下载的安装包解压缩到由步骤 2.2 产生的 `D:\Eclipse` 目录下，解压后的路径是 `D:\eclipse\GNU MCU Eclipse\Build Tools\`。
  - 虽然可以将下载的安装包解压到任何目录下，但是我们建议所有的工具都安装在某一个文件夹下，这样便于将来查找和操作，本文档描述的都安装在 `D:\eclipse` 目录下。
- 进入 *Eclipse* 菜单 → **Windows** → **Preferences** 将 *Global Build Tools Path* 设置成 Windows Build Tools 刚刚安装的路径，如图 Figure 1 所示。

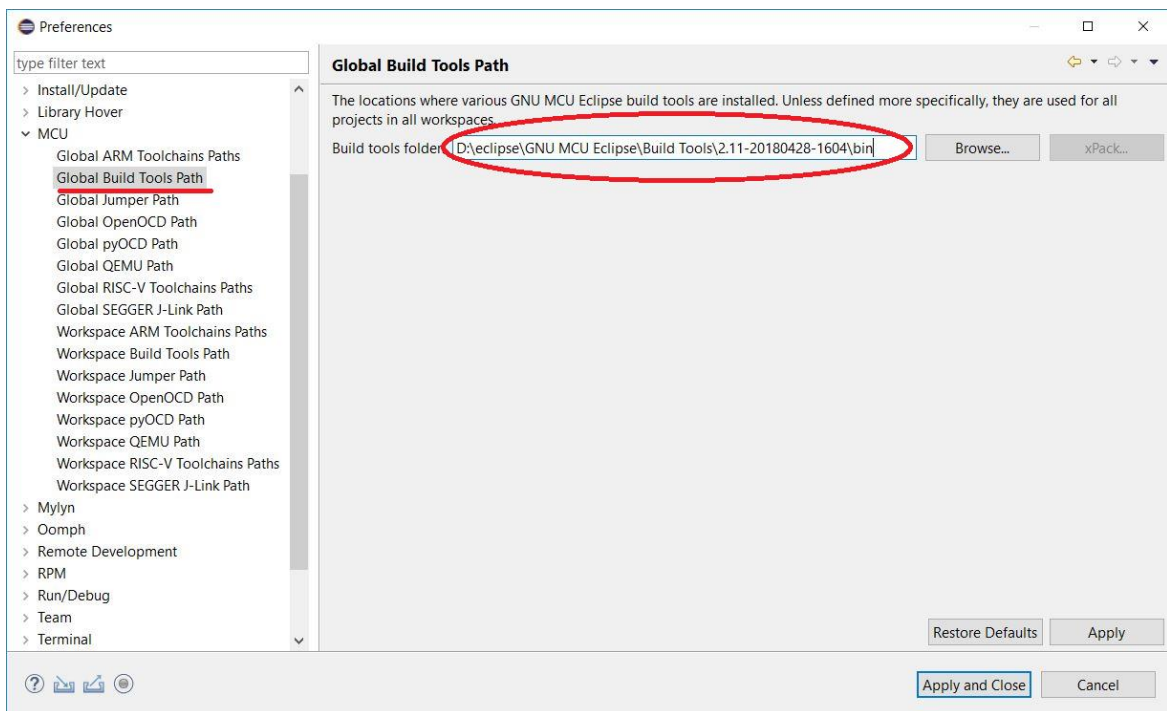


Figure 1. 设置 Global Build Tools Path

## 2.4 安装 RV32M1 GNU GCC 工具链

[www.open-isa.cn](http://www.open-isa.cn) 提供编译好的 RV32M1 的 GNU GCC 工具链可供下载。RV32M1 嵌入式系统的工具链包括 RISC-V C 和 C++ 交叉编译器，汇编器，连接器，binutils 工具包以及嵌入式 Newlib 库。其安装步骤如下：

- 从 [www.open-isa.cn](http://www.open-isa.cn) 下载编译好的 RV32M1 的 GNU GCC 工具链压缩包。
- 将下载的安装包解压缩到由步骤 2.3 产生的 `D:\eclipse\GNU MCU Eclipse` 目录下。
  - 虽然可以将下载的工具链安装包解压到任何目录下，但是我们建议将其放置在 `D:\eclipse\GNU MCU Eclipse` 下，这样便于将来查找和操作。
- 进入 *Eclipse* 菜单 **Windows** → **Preferences** 将 *Global RISC-V Toolchains Paths* 设置成工具链刚刚安装的路径，如图 Figure 2 所示。

4. 创建一个 Windows 环境变量 `RISCV32GCC_DIR` 并将其值设置为 GNU GCC Toolchain 的目录路径 `D:\eclipse\GNU MCU Eclipse\riscv32-unknown-elf-gcc`。
5. 将 GNU GCC 工具链的可执行文件所在的目录路径 `D:\eclipse\GNU MCU Eclipse\riscv32-unknown-elf-gcc\bin`，添加到用户或者系统的路径变量 `PATH` 中。

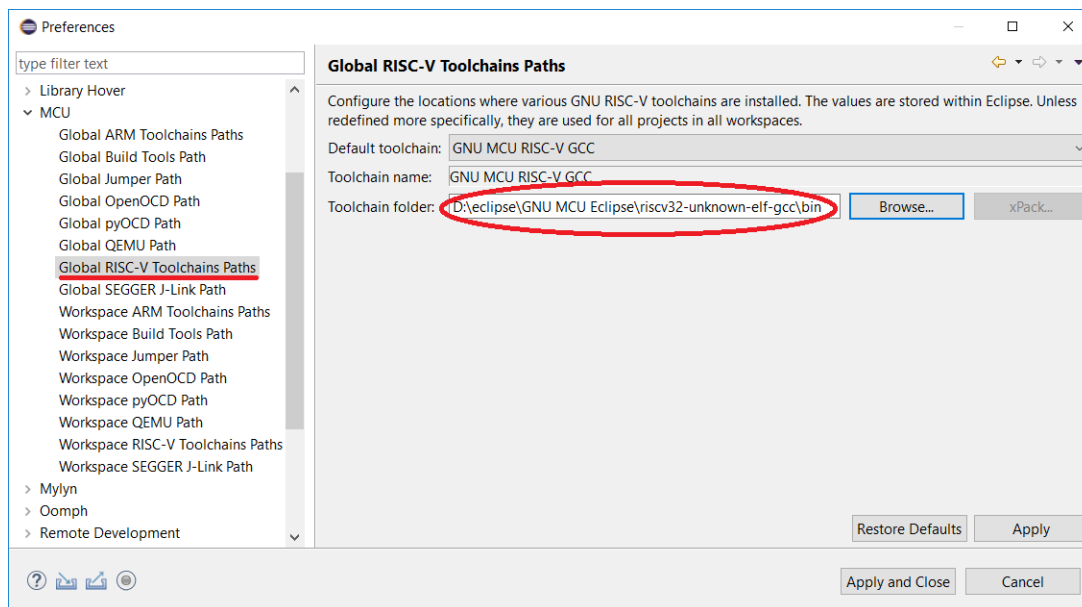


Figure 2. 设置 Global RISC-V Toolchains Paths

## 2.5 安装OpenOCD调试软件和J-Link驱动

[www.open-isa.cn](http://www.open-isa.cn) 提供有编译好的 OpenOCD 调试软件和 J-Link 驱动程序软件可供下载。按照下面的步骤按照 OpenOCD 调试软件：

1. 从 [www.open-isa.cn](http://www.open-isa.cn) 下载编译好的 OpenOCD 和 J-Link 驱动程序。
2. 将下载的安装包解压缩到由步骤 2.3 产生的 `D:\eclipse\GNU MCU Eclipse` 目录下。
  - 虽然可以将下载的工具链安装包解压到任何目录下，但是我们建议将其放置在 `D:\eclipse\GNU MCU Eclipse` 下，这样便于将来查找和操作。
3. 进入 *Eclipse* 菜单 **Windows** → **Preferences** 将 *Global OpenOCD Path* 设置成刚刚 OpenOCD 调试软件安装的路径，如图 Figure 3 所示。
4. 添加 OpenOCD.exe 的路径 `D:\eclipse\GNU MCU Eclipse\openocd\bin` 到环境变量 `PATH` 中。
5. 将 J-Link 硬件调试器连接到 Windows 电脑。
6. 运行 `D:\eclipse\GNU MCU Eclipse\openocd\drivers\drivers\UsbDriverTool.exe` 程序。
7. 右击 "J-link driver", 选择 "Install WinUSB" 安装 USB 驱动，如图 Figure 4 所示。

- 安装 USB 驱动后，J-Link 就不能再被 SEGGER 软件识别访问了。要恢复让 SEGGER 软件可以识别和访问 J-Link，可以运行 UsbDriverTool.exe 然后点击 "Restore default driver" 来恢复 J-Link 驱动，如图 Figure 5 所示。

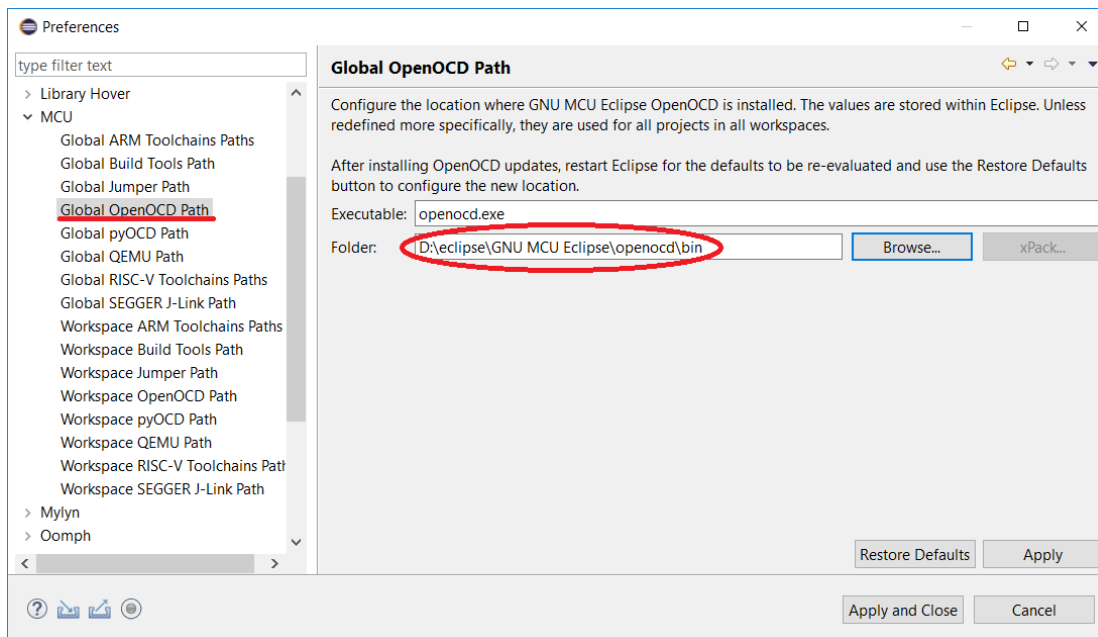


Figure 3. 设置 Global OpenOCD Path

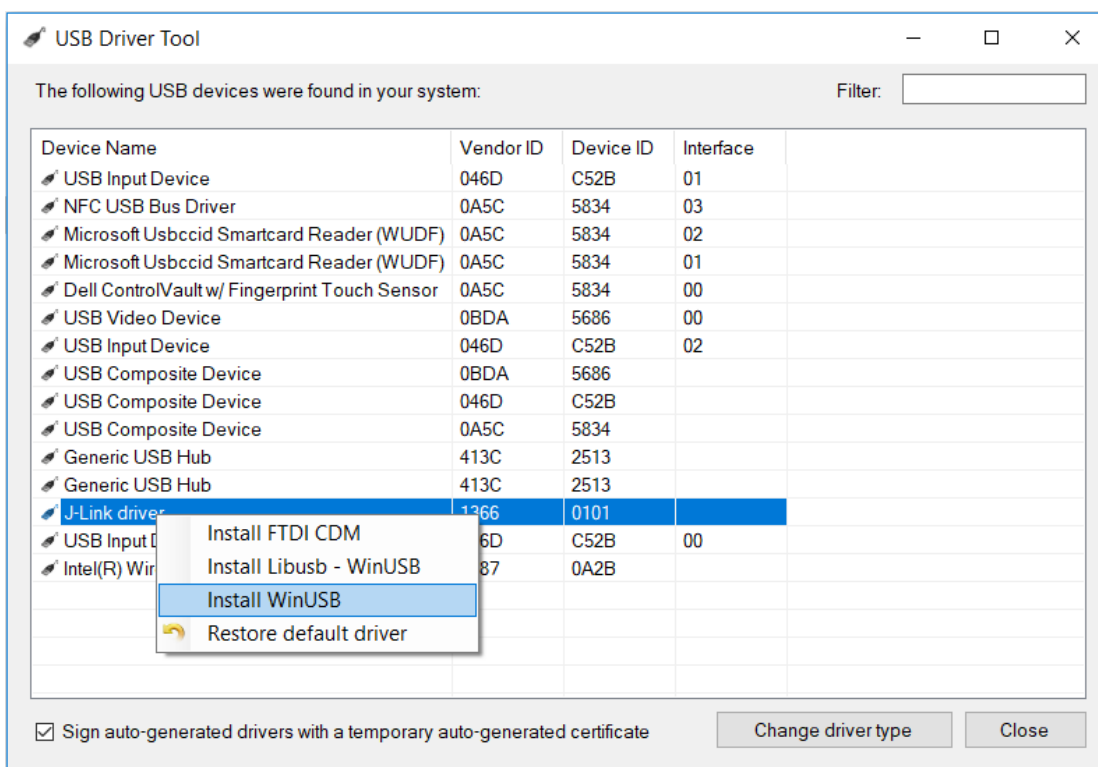


Figure 4. 安装 J-Link USB 驱动

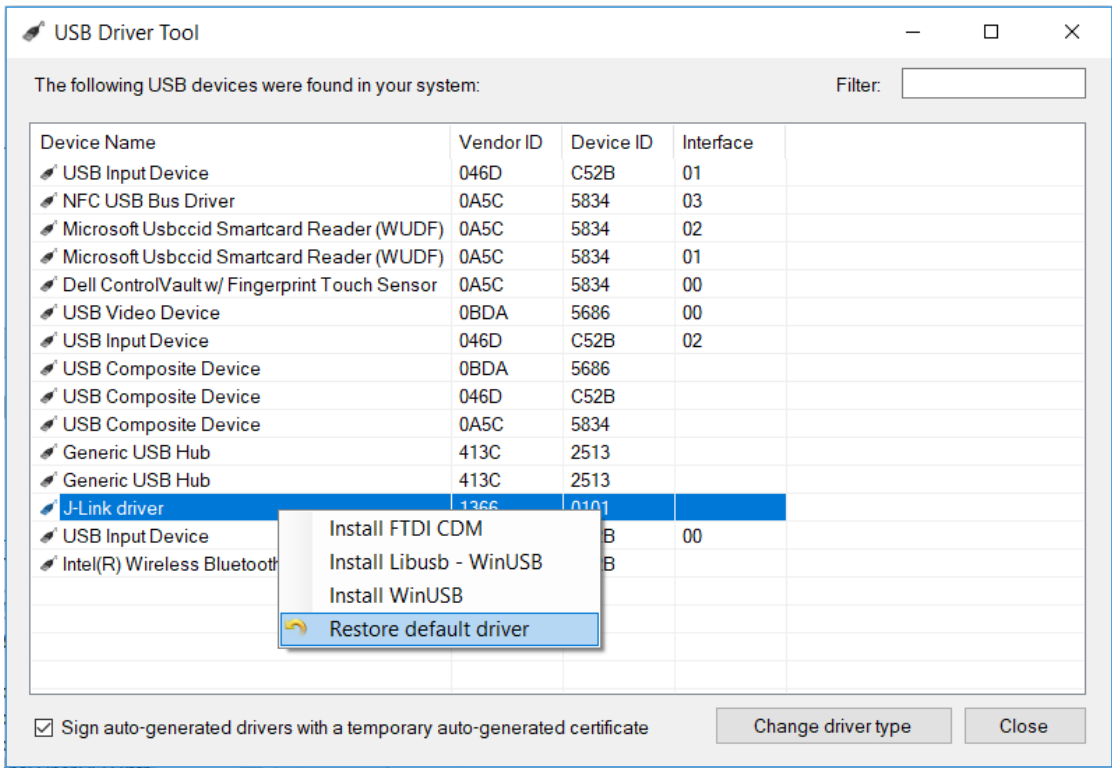


Figure 5. 恢复 J-Link USB 驱动

## 2.6 安装RV32M1 SDK

RV32M1 的 SDK 全部是以源代码的形式提供给用户下载。用户可以选择对应操作系统的 SDK 安装包下载安装。从 [www.open-isa.cn](http://www.open-isa.cn) 下载最新的 SDK 发布包到本地，然后解压下载的 SDK 压缩包到你希望的本地目录。

现在你可以使用 Eclipse IDE 来导入 RV32M1 SDK 例子工程，然后编译、调试、下载和运行这些例子程序了。

如果你习惯使用命令行的方式来编译、调试和运行程序，还需要安装一些其它的软件工具。这些工具使得命令行的编译调试环境更加便捷。接下来的内容将介绍如何安装这些的软件工具，

## 2.7 安装 CMake

从 <https://cmake.org/download/> 下载最新的 CMake 发布包。根据 Windows 系统版本选择下载 - win32-x86 或者 -win64-x64 版本的发布包。

建议使用 Windows 安装文件来安装 CMake 并确保在安装过程中选中 "Add CMake to system PATH" 选项。用户可以选择是否将 CMak 可执行路径安装到全部用户的环境变量 PATH 还是仅仅当前用户的环境变量 PATH。

如果用户下载的是 zip 文件然后解压手动安装 CMake 的，需要在安装后将其路径添加到环境变量 PATH 中去。



## 2.8 为命令行方式安装MinGW

MinGW 是一套最小的 GNU 工具集合，和 Cygwin 不同，它不依赖于第三方 C 运行时动态链接库。RV32M1 SDK 使用的编译环境没有使用 MinGW 的编译工具，而是利用 MinGW 提供的符合 GNU 的 GNU 工作环境，像类似 Unix/Linux 接口的基本 shell 控制台。MinGW 的安装步骤如下：

1. 从 [sourceforge.net/projects/mingw/files/Installer/](http://sourceforge.net/projects/mingw/files/Installer/) 下载最新版的 MinGW mingw-get-setup 安装文件
2. 运行下载的安装文件。建议的安装路径是 `C:\MinGW`，但是用户可以选择其它的安装路径。注意：安装路径不能包含任何空格。
3. 确保选中 **Basic Setup** 下面的“mingw32-base”和“msys-base”。如图 Figure 6 所示。

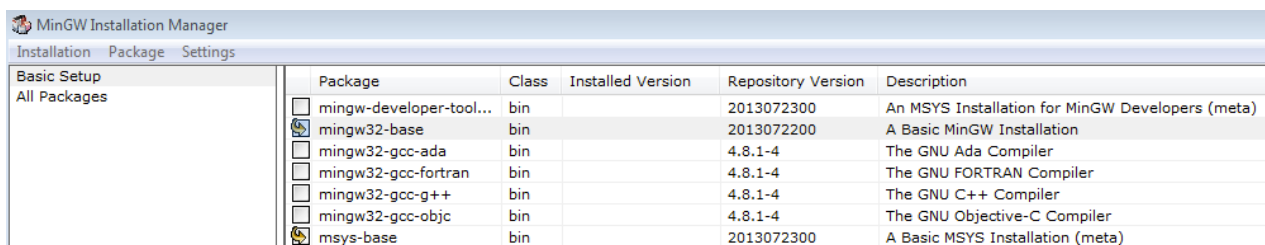


Figure 6. 安装 MinGW 和 MSYS

4. 在菜单“**Installation**”中选中“**Apply Changes**”，然后依照提示完成安装，如图 Figure 7 所示。

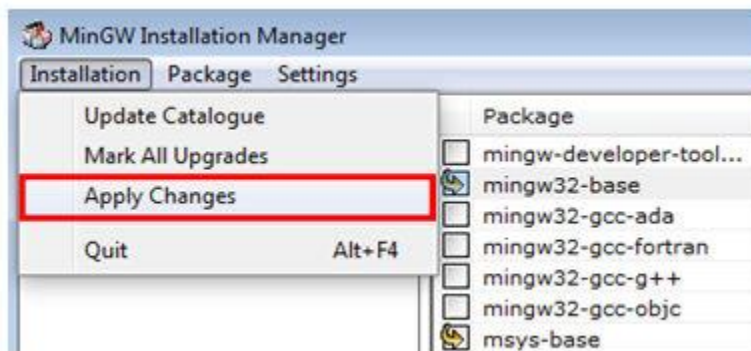


Figure 7. 完成 MinGW 和 MSYS 安装

5. 添加 `<mingw_install_dir>\bin` 到环境变量 PATH 中。

### 3. Linux下开发环境的搭建

本节介绍在 Linux 操作系统下为 RV32M1 搭建 RISC-V 开发环境，需要安装包括基于 RV32M1 SDK 开发、编译、调试和运行 RV32M1 程序所需要的 GNU GCC 工具链、OpenOCD 调试软件和 Eclipse 集成开发环境。所有的操作步骤是在 Ubuntu 18.04 测试过。

#### 3.1 安装RV32M1 GNU GCC 工具链和CMake

##### 1. 安装必需和依赖的软件（库）

```
sudo apt-get install autoconf automake autotools-dev curl libmpc-dev libmpfr-dev
libgmp-dev gawk build-essential bison flex texinfo gperf libtool patchutils bc
zlib1g-dev libusb-1.0-0-dev libudev libudev1 libudev-dev g++
```

##### 2. 克隆 PULP GNU 工具链源代码，打 patch 以支持 RV32M1，编译 RV32M1 GNU 工具链

```
# clone pulp-riscv-gnu-toolchain
git clone --recursive https://github.com/pulp-platform/pulp-riscv-gnu-toolchain
# clone RV32M1 GNU toolchain patch
git clone https://github.com/open-isa-rv32m1/rv32m1_gnu_toolchain_patch.git
# Copy and apply the patch
cp rv32m1_gnu_toolchain_patch/apply_rv32m1_patches.sh pulp-riscv-gnu-toolchain/
cp -r rv32m1_gnu_toolchain_patch/rv32m1_patches pulp-riscv-gnu-toolchain/
cd pulp-riscv-gnu-toolchain
bash ./apply_rv32m1_patches.sh
# Build the toolchain
./configure --prefix=/opt/pulp --with-arch=rv32imc --with-cmodel=medlow --enable-
multilib
sudo make
```

上面的--prefix 配置参数是将工具链安装到 /opt/pulp 目录。用户可以更改配置让工具链安装到任何自己需要位置。

[www.open-isa.cn](http://www.open-isa.cn) 提供编译好的 RV32M1 的 GNU GCC 工具链可供下载。用户如果不想自己编译可以直接下载安装。

##### 3. 安装 CMake

```
sudo apt-get install cmake
```



- 更新系统变量。如果用户修改默认的工具链安装位置，`RISCV32GCC_DIR` 要设置成修改的位置。

```
export PATH=$PATH:/opt/pulp/bin
export RISCV32GCC_DIR="/opt/pulp"
```

## 3.2 安装OpenOCD调试软件

下面的安装步骤是在 Ubuntu 18.04 测试过。

- 安装依赖的软件（库）

```
sudo apt-get install autoconf automake autotools-dev curl libmpc-dev libmpfr-dev
libgmp-dev gawk build-essential bison flex texinfo gperf libtool patchutils bc
zlib1g-dev libusb-1.0-0-dev libudev libudev1 libudev-dev g++
```

- 克隆和编译 OpenOCD 调试软件

```
# Download the source code
git clone --recurse https://github.com/open-isa-rv32m1/rv32m1-openocd.git
# Build and install
cd rv32m1-openocd
./bootstrap
./configure
make
sudo make install
```

[www.open-isa.cn](http://www.open-isa.cn) 也提供编译好的 OpenOCD 可供下载。用户如果不想自己编译可以直接下载安装，并确保安装位置添加到系统变量 `PATH` 中去。

## 3.3 安装Eclipse IDE for C/C++ Developers

安装下面的步骤安装 Eclipse IDE for C/C++ 集成开发环境：

- 从 [Github](https://github.com) 下载 GNU MCU Eclipse IDE for C/C++。
  - GNU MCU Eclipse IDE for C/C++ 包括了 Eclipse CDT 和 the MCU plug-ins。
  - 选择最新版的下载。目前最新版的不再支持 32 位操作系统。
  - 选择文件名后面带有 `-linux.gtk.x86_64.tar.gz` 字样的是 Linux 版本的安装包
- 将下载的压缩包解压并移动其 Eclipse 文件夹到你希望的位置。
- 根据 [2.4](#) 和 [2.5](#) 小节中的方法，设置 *Global RISC-V Toolchains Paths* 和 *Global OpenOCD Path* 为对应的目录位置。

## 4. MacOS下开发环境的搭建

本节介绍在 MacOS 操作系统下为 RV32M1 搭建 RISC-V 开发环境，需要安装包括基于 RV32M1 SDK 开发、编译、调试和运行 RV32M1 程序所需要的 GNU GCC 工具链、OpenOCD 调试软件和 Eclipse 集成开发环境。

### 4.1 安装RV32M1 GNU GCC 工具链和CMake

#### 1. 安装必需和依赖的软件（库）

```
brew install gawk gnu-sed gmp mpfr libmpc isl zlib
```

#### 2. 克隆 PULP GNU 工具链源代码，打 patch 以支持 RV32M1，编译 RV32M1 GNU 工具链

```
# clone pulp-riscv-gnu-toolchain
git clone --recursive https://github.com/pulp-platform/pulp-riscv-gnu-toolchain
# clone RV32M1 GNU toolchain patch
git clone https://github.com/open-isa-rv32m1/rv32m1_gnu_toolchain_patch.git
# Copy and apply the patch
cp rv32m1_gnu_toolchain_patch/apply_rv32m1_patches.sh pulp-riscv-gnu-toolchain/
cp -r rv32m1_gnu_toolchain_patch/rv32m1_patches pulp-riscv-gnu-toolchain/
cd pulp-riscv-gnu-toolchain
bash ./apply_rv32m1_patches.sh
# Build the toolchain
./configure --prefix=/opt/pulp --with-arch=rv32imc --with-cmodel=medlow --enable-multilib
make
```

上面的--prefix 配置参数是将工具链安装到 /opt/pulp 目录。用户可以配置安装到任何自己需要位置。

[www.open-isa.cn](http://www.open-isa.cn) 提供编译好的 RV32M1 的 GNU GCC 工具链可供下载。用户如果不想自己编译可以直接下载安装。

#### 3. 安装 CMake

```
brew install cmake
```

#### 4. 更新系统变量。如果用户修改默认的安装位置，RISCV32GCC\_DIR 要设置成修改的位置

```
export PATH=$PATH:/opt/pulp/bin
export RISCV32GCC_DIR="/opt/pulp"
```

## 4.2 安装OpenOCD

1. 在 MacOS 下编译安装和运行 OpenOCD 需要安装 Xcode5、命令行工具（可以从 **Xcode5->Preferences->Downloads** 菜单下安装）和 Homebrew 工具。然后运行

```
brew install libtool automake libusb libusb-compat
```

2. 克隆和编译 OpenOCD 调试软件

```
# Download the source code
git clone --recurse https://github.com/open-isa-rv32m1/rv32m1-openocd.git
# Build and install
cd rv32m1-openocd
./bootstrap
./configure
make
sudo make install
```

[www.open-isa.cn](http://www.open-isa.cn) 也提供编译好的 OpenOCD 可供下载。用户如果不想自己编译可以直接下载安装，并确保安装位置添加到系统变量 PATH 中去。

## 4.3 安装Eclipse IDE for C/C++ Developers

安装下面的步骤安装 Eclipse IDE for C/C++ 集成开发环境：

4. 从 [Github](https://github.com) 下载 GNU MCU Eclipse IDE for C/C++。
  - GNU MCU Eclipse IDE for C/C++ 包括了 Eclipse CDT 和 the MCU plug-ins。
  - 选择最新版的下载。目前最新版的不再支持 32 位操作系统。
  - 选择文件名后面带有 **-macosx.cocoa.x86\_64.tar.gz** 字样的是 MacOS 版本的安装包
5. 将下载的压缩包解压并移动其 Eclipse 文件夹到你希望的位置。
6. 根据 [2.4](#) 和 [2.5](#) 小节中的方法，设置 *Global RISC-V Toolchains Paths* 和 *Global OpenOCD Path* 为对应的目录位置。

## 5 参考文档

可以从 [www.open-isa.cn](http://www.open-isa.cn) 下载以下参考文档：

- VEGA-Lite-SCH: 原理图
- VEGA-Lite-LAYOUT: PCB 元器件分布图
- 织女星开发板硬件用户指南. pdf
- RV32M1RM: 芯片参考手册
- RV32M1DS: 芯片数据手册

## 6 修订历史

版本号	日期	修改内容
0	04/2019	第一版



VEGA\*